

Numerical Simulation of Dynamic Systems: Hw6 - Solution

Prof. Dr. François E. Cellier
Department of Computer Science
ETH Zurich

April 3, 2012

[H5.3] Stability Domain of GE4/AB3

The method introduced in earlier chapters for drawing stability domains was geared towards *linear time-invariant homogeneous multi-variable state-space models*:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$$

We generated real-valued \mathbf{A} -matrices $\in \mathbb{R}^{2 \times 2}$ with their eigenvalues located on the unit circle, at an angle α away from the negative real axis. We then computed the \mathbf{F} -matrix corresponding to that \mathbf{A} -matrix for the given algorithm, and found the largest value of the step size h , for which all eigenvalues of \mathbf{F} remained inside the unit circle. This gave us one point on the stability domain. We repeated this procedure for all suitable values of the angle α .

[H5.3] Stability Domain of GE4/AB3

The method introduced in earlier chapters for drawing stability domains was geared towards *linear time-invariant homogeneous multi-variable state-space models*:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x}$$

We generated real-valued \mathbf{A} -matrices $\in \mathbb{R}^{2 \times 2}$ with their eigenvalues located on the unit circle, at an angle α away from the negative real axis. We then computed the \mathbf{F} -matrix corresponding to that \mathbf{A} -matrix for the given algorithm, and found the largest value of the step size h , for which all eigenvalues of \mathbf{F} remained inside the unit circle. This gave us one point on the stability domain. We repeated this procedure for all suitable values of the angle α .

The algorithm needs to be modified for dealing with second derivative systems described by the *linear time-invariant homogeneous multi-variable second-derivative model*:

$$\ddot{\mathbf{x}} = \mathbf{A}^2 \cdot \mathbf{x} + \mathbf{B} \cdot \dot{\mathbf{x}}$$

[H5.3] Stability Domain of GE4/AB3 II

We need to find real-valued **A**- and **B**-matrices such that the second derivative model has its eigenvalues located on the unit circle.

[H5.3] Stability Domain of GE4/AB3 II

We need to find real-valued **A**- and **B**-matrices such that the second derivative model has its eigenvalues located on the unit circle.

This can be accomplished using the scalar model:

$$\ddot{x} = a^2 \cdot x + b \cdot \dot{x}$$

where:

$$\begin{aligned} a &= \sqrt{a_{21}} \\ b &= a_{22} \end{aligned}$$

of the formerly used **A**-matrix.

[H5.3] Stability Domain of GE4/AB3 III

Write the GE4/AB3 algorithm as follows:

$$\begin{aligned}
 x_{k+1} &= \frac{20}{11} \cdot x_k - \frac{6}{11} \cdot x_{k-1} - \frac{4}{11} \cdot x_{k-2} + \frac{1}{11} \cdot x_{k-3} + \frac{12 \cdot h^2}{11} \cdot \ddot{x}_k \\
 h \cdot \dot{x}_{k+1} &= h \cdot \dot{x}_k + \frac{23 \cdot h^2}{12} \cdot \ddot{x}_k - \frac{4 \cdot h^2}{3} \cdot \ddot{x}_{k-1} + \frac{5 \cdot h^2}{12} \cdot \ddot{x}_{k-2} \\
 \ddot{x} &= a^2 \cdot x + b \cdot \dot{x}
 \end{aligned}$$

Substitute the model equation into the two solver equations, and rewrite the resulting equations in a state-space form:

$$\xi_{k+1} = \mathbf{F} \cdot \xi_k$$

[H5.3] Stability Domain of GE4/AB3 IV

whereby the state vector ξ is chosen as:

$$\xi_k = \begin{pmatrix} x_{k-3} \\ h \cdot \dot{x}_{k-3} \\ x_{k-2} \\ h \cdot \dot{x}_{k-2} \\ x_{k-1} \\ h \cdot \dot{x}_{k-1} \\ x_k \\ h \cdot \dot{x}_k \end{pmatrix}$$

The F -matrix turns out to be a function of $(a \cdot h)^2$ and of $b \cdot h$.

[H5.3] Stability Domain of GE4/AB3 IV

whereby the state vector ξ is chosen as:

$$\xi_k = \begin{pmatrix} x_{k-3} \\ h \cdot \dot{x}_{k-3} \\ x_{k-2} \\ h \cdot \dot{x}_{k-2} \\ x_{k-1} \\ h \cdot \dot{x}_{k-1} \\ x_k \\ h \cdot \dot{x}_k \end{pmatrix}$$

The **F**-matrix turns out to be a function of $(a \cdot h)^2$ and of $b \cdot h$.

The remainder of the algorithm remains the same as before.

[H5.3] Stability Domain of GE4/AB3 IV

whereby the state vector ξ is chosen as:

$$\xi_k = \begin{pmatrix} x_{k-3} \\ h \cdot \dot{x}_{k-3} \\ x_{k-2} \\ h \cdot \dot{x}_{k-2} \\ x_{k-1} \\ h \cdot \dot{x}_{k-1} \\ x_k \\ h \cdot \dot{x}_k \end{pmatrix}$$

The F -matrix turns out to be a function of $(a \cdot h)^2$ and of $b \cdot h$.

The remainder of the algorithm remains the same as before.

Draw the stability domain of GE4/AB3 using this approach.

[H5.3] Stability Domain of GE4/AB3 V

The **F**-matrix of **GE4/AB3** is:

$$\mathbf{F} = \begin{pmatrix} \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} \\ -\frac{1}{11}\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & -\frac{4}{11}\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & -\frac{6}{11}\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \left[\frac{20}{11}\mathbf{I}^{(n)} + \frac{12}{11}(\mathbf{A}h)^2\right] & \frac{12}{11}(\mathbf{B}h) \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \frac{5}{12}(\mathbf{A}h)^2 & \frac{5}{12}(\mathbf{B}h) & -\frac{4}{12}(\mathbf{A}h)^2 & -\frac{4}{12}(\mathbf{B}h) & \frac{23}{12}(\mathbf{A}h)^2 & \left[\mathbf{I}^{(n)} + \frac{23}{12}(\mathbf{B}h)\right] \end{pmatrix}$$

[H5.3] Stability Domain of GE4/AB3 V

The **F**-matrix of **GE4/AB3** is:

$$F = \begin{pmatrix} Z^{(n)} & Z^{(n)} & I^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} \\ Z^{(n)} & Z^{(n)} & Z^{(n)} & I^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} \\ Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & I^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} \\ Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & I^{(n)} & Z^{(n)} & Z^{(n)} \\ Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & I^{(n)} & Z^{(n)} \\ Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & Z^{(n)} & I^{(n)} \\ -\frac{1}{11}I^{(n)} & Z^{(n)} & -\frac{4}{11}I^{(n)} & Z^{(n)} & -\frac{6}{11}I^{(n)} & Z^{(n)} & \left[\frac{20}{11}I^{(n)} + \frac{12}{11}(Ah)^2 \right] & \frac{12}{11}(Bh) \\ Z^{(n)} & Z^{(n)} & \frac{5}{12}(Ah)^2 & \frac{5}{12}(Bh) & -\frac{4}{12}(Ah)^2 & -\frac{4}{12}(Bh) & \frac{23}{12}(Ah)^2 & \left[I^{(n)} + \frac{23}{12}(Bh) \right] \end{pmatrix}$$

We generate the **A**-matrix as always, then extract:

$$a = a_{21} \cdot h^2$$

$$b = a_{22} \cdot h$$

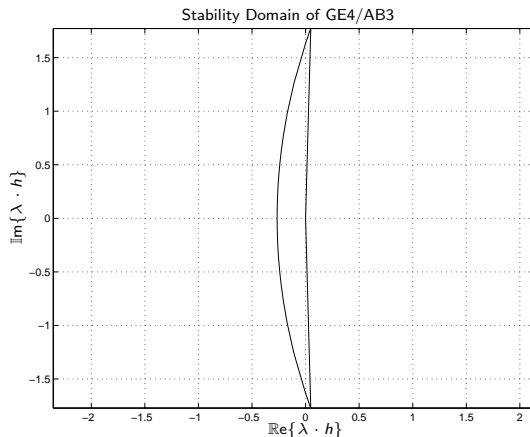
[H5.3] Stability Domain of GE4/AB3 VI

We then rewrite the **F**-matrix as follows:

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{1}{11} & 0 & -\frac{4}{11} & 0 & -\frac{6}{11} & 0 & \left[\frac{20}{11} + \frac{12}{11}a\right] & \frac{12}{11}b \\ 0 & 0 & \frac{5}{12}a & \frac{5}{12}b & -\frac{4}{12}a & -\frac{4}{12}b & \frac{23}{12}a & \left[1 + \frac{23}{12}b\right] \end{pmatrix}$$

[H5.3] Stability Domain of GE4/AB3 VII

We can now apply our standard stability domain plotting routine.



[P5.1] Houbolt's Integration Algorithm

John Houbolt proposed already in 1950 a second-derivative integration algorithm that is very similar to the GL3/BDF2 method introduced in this chapter. *Houbolt's algorithm* can be written as follows:

$$\begin{aligned} \mathbf{x}_{k+1} &= \frac{5}{2} \cdot \mathbf{x}_k - 2 \cdot \mathbf{x}_{k-1} + \frac{1}{2} \cdot \mathbf{x}_{k-2} + \frac{h^2}{2} \cdot \ddot{\mathbf{x}}_{k+1} \\ h \cdot \dot{\mathbf{x}}_{k+1} &= \frac{11}{6} \cdot \mathbf{x}_{k+1} - 3 \cdot \mathbf{x}_k + \frac{3}{2} \cdot \mathbf{x}_{k-1} - \frac{1}{3} \cdot \mathbf{x}_{k-2} \end{aligned}$$

[P5.1] Houbolt's Integration Algorithm

John Houbolt proposed already in 1950 a second-derivative integration algorithm that is very similar to the GI3/BDF2 method introduced in this chapter. *Houbolt's algorithm* can be written as follows:

$$\begin{aligned} \mathbf{x}_{k+1} &= \frac{5}{2} \cdot \mathbf{x}_k - 2 \cdot \mathbf{x}_{k-1} + \frac{1}{2} \cdot \mathbf{x}_{k-2} + \frac{h^2}{2} \cdot \ddot{\mathbf{x}}_{k+1} \\ h \cdot \dot{\mathbf{x}}_{k+1} &= \frac{11}{6} \cdot \mathbf{x}_{k+1} - 3 \cdot \mathbf{x}_k + \frac{3}{2} \cdot \mathbf{x}_{k-1} - \frac{1}{3} \cdot \mathbf{x}_{k-2} \end{aligned}$$

The second derivative formula of Houbolt's algorithm can immediately be identified as **GI3**. The formula used for the velocity vector is **BDF3**; however, the formula was used differently from the way, it had been employed by us in the description of the GI3/BDF2 algorithm. Clearly, the Houbolt algorithm is third-order accurate. Although it would have sufficed to use BDF2 for the velocity vector, nothing would have been gained computationally by choosing the reduced-order algorithm.

[P5.1] Houbolt's Integration Algorithm II

We can transform the Houbolt algorithm to the form that we meanwhile got used to by substituting the G13 solver into the BDF3 solver to eliminate \mathbf{x}_{k+1} from the latter. The so rewritten Houbolt algorithm assumes the form:

$$\begin{aligned}\mathbf{x}_{k+1} &= \frac{5}{2} \cdot \mathbf{x}_k - 2 \cdot \mathbf{x}_{k-1} + \frac{1}{2} \cdot \mathbf{x}_{k-2} + \frac{h^2}{2} \cdot \ddot{\mathbf{x}}_{k+1} \\ h \cdot \dot{\mathbf{x}}_{k+1} &= \frac{19}{12} \cdot \mathbf{x}_k - \frac{13}{6} \cdot \mathbf{x}_{k-1} + \frac{7}{12} \cdot \mathbf{x}_{k-2} + \frac{11 \cdot h^2}{12} \cdot \ddot{\mathbf{x}}_{k+1}\end{aligned}$$

Find the stability domain and damping plot of Houbolt's algorithm, and discuss the properties of this algorithm in the same way, as Newmark's algorithm has been discussed in class.

[P5.1] Houbolt's Integration Algorithm III

The **F**-matrix of the *Houbolt algorithm* is:

$$\mathbf{F} = \begin{pmatrix} \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} \\ \frac{1}{2}\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & -2\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \left[\frac{5}{2}\mathbf{I}^{(n)} + \frac{1}{2}(\mathbf{A}h)^2\right] & \frac{1}{2}(\mathbf{B}h) \\ \frac{7}{12}\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & -\frac{13}{6}\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \left[\frac{19}{12}\mathbf{I}^{(n)} + \frac{11}{12}(\mathbf{A}h)^2\right] & \frac{11}{12}(\mathbf{B}h) \end{pmatrix}$$

[P5.1] Houbolt's Integration Algorithm III

The **F**-matrix of the *Houbolt algorithm* is:

$$\mathbf{F} = \begin{pmatrix} \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} & \mathbf{Z}^{(n)} \\ \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{Z}^{(n)} & \mathbf{I}^{(n)} \\ \frac{1}{2}\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & -2\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \left[\frac{5}{2}\mathbf{I}^{(n)} + \frac{1}{2}(\mathbf{A}h)^2\right] & \frac{1}{2}(\mathbf{B}h) \\ \frac{7}{12}\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & -\frac{13}{6}\mathbf{I}^{(n)} & \mathbf{Z}^{(n)} & \left[\frac{19}{12}\mathbf{I}^{(n)} + \frac{11}{12}(\mathbf{A}h)^2\right] & \frac{11}{12}(\mathbf{B}h) \end{pmatrix}$$

We generate the **A**-matrix as always, then extract:

$$\begin{aligned} a &= a_{21} \cdot h^2 \\ b &= a_{22} \cdot h \end{aligned}$$

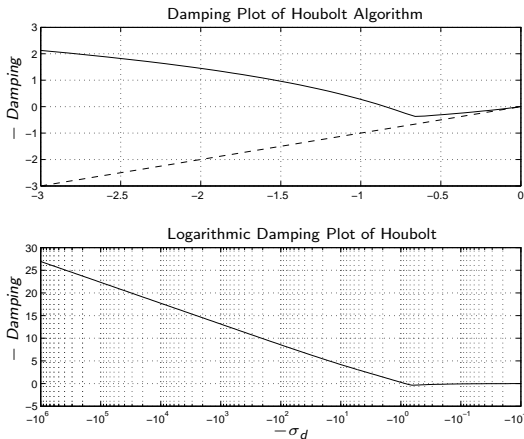
[P5.1] Houbolt's Integration Algorithm IV

We then rewrite the **F**-matrix as follows:

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{2} & 0 & -2 & 0 & \left[\frac{5}{2} + \frac{1}{2}a\right] & \frac{1}{2}b \\ \frac{7}{12} & 0 & -\frac{13}{6} & 0 & \left[\frac{19}{12} + \frac{11}{12}a\right] & \frac{11}{12}b \end{pmatrix}$$

[P5.1] Houbolt's Integration Algorithm V

Let us start by drawing the *damping plot* of the Houbolt algorithm:



[P5.1] Houbolt's Integration Algorithm VI

- ▶ Houbolt's algorithm exhibits negative damping farther out along the negative real axis. Thus, the algorithm is *not stiffly stable*, and the stability domain will loop in the left-half complex plane.

[P5.1] Houbolt's Integration Algorithm VI

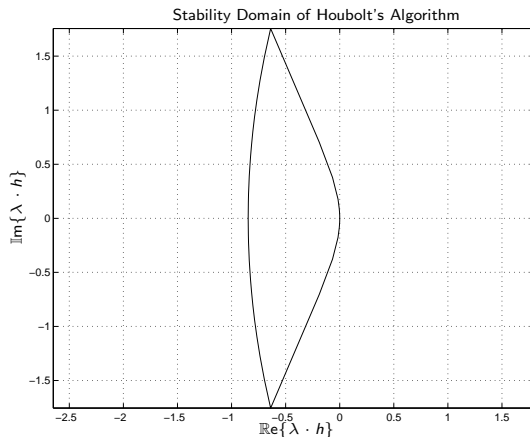
- ▶ Houbolt's algorithm exhibits negative damping farther out along the negative real axis. Thus, the algorithm is *not stiffly stable*, and the stability domain will loop in the left-half complex plane.
- ▶ Houbolt's algorithm has an asymptotic region around the origin, which, unfortunately, is rather small. Along the negative real axis, the asymptotic region ends for $\sigma_d \approx 0.1$.

[P5.1] Houbolt's Integration Algorithm VI

- ▶ Houbolt's algorithm exhibits negative damping farther out along the negative real axis. Thus, the algorithm is *not stiffly stable*, and the stability domain will loop in the left-half complex plane.
- ▶ Houbolt's algorithm has an asymptotic region around the origin, which, unfortunately, is rather small. Along the negative real axis, the asymptotic region ends for $\sigma_d \approx 0.1$.
- ▶ Since Houbolt's algorithm is an *implicit second-derivative ODE solver*, we had hoped for stiff stability. Without that property, the algorithm will perform poorly in comparison with the explicit GE3/AB2 algorithm.

[P5.1] Houbolt's Integration Algorithm VII

Let us now plot the stability domain of Houbolt's algorithm:



[P5.1] Houbolt's Integration Algorithm VIII

- ▶ The flying-saucer stability domain of Houbolt's algorithm looks beautiful, but the method is hardly convincing.

[P5.1] Houbolt's Integration Algorithm VIII

- ▶ The flying-saucer stability domain of Houbolt's algorithm looks beautiful, but the method is hardly convincing.
- ▶ Unfortunately, we still haven't discovered an *L-stable second-derivative ODE solver*.

[P5.1] Houbolt's Integration Algorithm VIII

- ▶ The flying-saucer stability domain of Houbolt's algorithm looks beautiful, but the method is hardly convincing.
- ▶ Unfortunately, we still haven't discovered an *L-stable second-derivative ODE solver*.
- ▶ Such an algorithm does probably exist, but we haven't found it yet.